

REMARKS

Claims 1-21 are pending in the Application. Claims 1-21 are rejected under 35 U.S.C. §102(e). Applicants respectfully traverse these rejections for at least the reasons stated below and respectfully request the Examiner to reconsider and withdraw these rejections.

I. EXAMINER'S RESPONSE TO ARGUMENTS

In an attempt to obtain clarification of the Examiner's response, Applicants' representative held a phone conference with the Examiner on 12/19/2005. In particular, the Examiner was asked to clarify her interpretation of two terms, "a log file parser" and "a database file." During this interview, the Examiner did not consult the cited reference *Singer* and was unable to provide any further details to the argument against independent claim 1, which contains these terms. In particular, the Examiner maintained that the collection server of *Singer* did perform log file parsing and deferred to what was already stated in the final rejection with respect to log file parsing. *See* Office Action from 11/30/2005, p. 5, line 12 to p. 6, line 2.

The Examiner states in reference to language used in claims 1, 8, and 15 in the Response to Arguments that:

*Singer* discloses a collection server (parser) that starts a batch program that determines the web servers at the location of the collection server and creates the collection program that collects the server logs from the web servers, zips the files (output files), and transfers the files (output files) to be stored in a database (paragraph [0054]). The collection server (log file parser) starts a batch and collection program on each server as the same physical location as it (paragraph [0054]). *See* Office Action from 11/30/2005, p. 5, lines 12-21.

Applicants respectfully traverse the Examiner's interpretation of the reference. *Singer* does not teach "starting a log file parser on each server of a set of servers in a distributed information processing environment." The collection server in *Singer* simply compresses and collects log files from a list of identified web servers. *Singer*

¶[0054]; Fig. 4, element 401, 402. The collection server then transfers the compressed log files to an analysis server. *Singer* ¶[0054], Fig. 4, element 403. At the analysis server, all log files are decompressed and then centrally parsed for loading into a database. *Singer* ¶[0057-0065]. There is no language or figure in *Singer* that teaches or suggests "starting a log file parser on each server of a set of servers in a distributed information processing environment," and "retrieving usage information from a database file generated by said log file parser." The specification and drawings establish the detailed description of a log file parser. See Specification, p. 11, lines 11-28, Fig. 4. In Figure 4, steps 403-404, a current log file is closed and a new log file is opened; in step 406 the log file closed in step 402 is read in and parsed, such that in step 408 a database file as output is generated. The term "parsed" is well known and consistently used in the art.

**parse** - Computer Science. To analyze or separate (input, for example) into more easily processed components. See The American Heritage® Dictionary of the English Language, Fourth Edition Copyright © 2000 by Houghton Mifflin Company.

**parsing** - In computer science, parsing is the process of analyzing a continuous stream of input (read from a file or a keyboard, for example) in order to determine its grammatical structure with respect to a given formal grammar. A parser is a computer program that carries out this task. The name is by analogy with the usage in grammar and linguistics. The term parseable is generally applied to text or data which can be parsed. Parsing transforms input text into a data structure, usually a tree, which is suitable for later processing and which captures the implied hierarchy of the input. Generally, parsers operate in two stages, first identifying the meaningful tokens in the input, and then building a parse tree from those tokens. See <http://en.wikipedia.org/wiki/Parser>.

**parse** - In linguistics, to divide language into small components that can be analyzed. For example, parsing this sentence would involve dividing it into words and phrases and identifying the type of each component (e.g., verb, adjective, or noun). Parsing is a very important part of many computer science disciplines. For example, compilers must parse source code to be able to translate it into object code. Likewise, any application that processes complex commands must be able to parse the commands. This includes virtually all end-user applications. Parsing is often divided into lexical analysis and semantic parsing. Lexical analysis concentrates on dividing strings into components, called tokens, based on punctuation and other keys. Semantic parsing then attempts to determine the meaning of the string. See <http://www.webopedia.com/TERM/P/parse.html>.

The collection server of *Singer* does not performing parsing, because the log files in *Singer* are parsed after transmission to and decompression by the analysis server.

*Singer* ¶[0058-0061]. Figure 4 of Applicants' disclosure clearly indicates that a log file parser is not anticipated or suggested by the collection server in *Singer*. *Singer* admits that the network architecture and location of the processing elements affects the speed, ergo efficiency, of the method. *Singer* ¶[0043]. By "starting a log file parser on each server of a set of servers" as in claims 1, 8, and 15, the method of the present invention can provide a unique distribution of computing functions and so is patentably distinct from the method of *Singer* and from the prior art.

Further, regarding claims 1, 8, and 15, the Examiner states in reference to language used in these claims in the Response to Arguments that:

*Singer* discloses the analysis server accessing the zipped files from the collections server by unzipping the log files, filtering the files for determined web server data, and loading the data into the database (paragraph [0057-0060]). The files contain usage data from the web servers (paragraph [0048, 0051]). The analysis server uses the files to generate summary tables and reports (paragraphs [0066, 0069]). Therefore, *Singer*, undoubtedly, discloses "retrieving usage information from a database file generated by said log file parser". See Office Action from 11/30/2005, p. 6, lines 3-9.

Applicants respectfully traverse the Examiner's interpretation of the reference. *Singer* does not teach "retrieving usage information from a database file generated by said log file parser." Since, as argued previously, *Singer* does not anticipate or suggest "said log file parser" which operates on "each server", the database file in *Singer* does not anticipate and is patentably distinct from "a database file" as in claims 1, 8, and 15. Further, the method of *Singer* does not anticipate retrieving usage information from a database file generated in output by said log file parser. The database files in *Singer* are clearly generated by parsing the log files at the analysis server, after transmission of the compressed log files by the collection server. *Singer* ¶[0057-0061].

As a result of the foregoing, Applicants respectfully request that the final rejection be reconsidered, and assert, in view of the arguments presented in support of claims 1-21, that the rejection of these claims be withdrawn.

## II. REJECTIONS UNDER 35 U.S.C. §102(e):

The Examiner has rejected claims 1-21 under 35 U.S.C. §102(e) as being anticipated by *Singer et al.* (U.S. Patent Publication No. 2005/0125531 A1), hereafter referred to as *Singer*. Applicants respectfully traverse these rejections for at least the reasons stated below and respectfully request the Examiner to reconsider and withdraw these rejections.

For a claim to be anticipated under 35 U.S.C. §102, each and every claim limitation must be found within the cited prior art reference and arranged as required by the claim. M.P.E.P. §2131.

Applicants respectfully assert that *Singer* does not disclose "starting a log file parser on each server of a set of servers in a distributed information processing environment" and "retrieving usage information from a database file generated by said log file parser" as recited in the method of claim 1, the computer program product of claim 8, and the data processing system of claim 15. Regarding the first step, the Examiner cites ¶[0008-0009, 0054] of *Singer* as disclosing the above-cited claim limitations. *See* Office Action from 11/30/2005, p. 2, lines 23-25. Applicants respectfully traverse. *Singer* does disclose a "system for collecting, filtering, analyzing and reporting web server usage data" (*Singer* ¶[0008]), and a process by which "web server usage data can be collected from multiple servers" (*Singer* ¶[0009]). However, the collection process of *Singer* does not contain the elements in claims 1, 8, and 15 of starting a log file parser on each server and generating a database file by said log file parser. The method disclosed by *Singer* begins with a pre-installed filtering program on each web server for locally generating a customized, filtered log file. *Singer* ¶[0051]. In the method of *Singer*, the Collection Server then starts a Collection Program that locally compresses the filtered log files (using WinZip V 6.3) and transfers these zipped files to the Analysis Server. *Singer* ¶[0054]. Hence, *Singer* discloses the collection, transfer, and management of filtered

log files. There is no language in *Singer* regarding generating a database file, on each server by a log file parser. Thus, as previously argued in detail in this paper in reference to the Examiner's response, *Singer* does not disclose all of the limitations of claims 1, 8, and 15 and thus *Singer* does not anticipate claims 1, 8, and 15. M.P.E.P. §2131.

Applicants further assert that *Singer* does not disclose "closing a current log file", "reading said log file", and "generating said database file in response to said log file," as recited in the method of claim 2, the computer program product of claim 9, and the data processing system of claim 16. The Examiner cites ¶¶0054, 0057-0060, 0061-0066] of *Singer* as disclosing the above-cited claim limitations. See Office Action from 11/30/2005, p. 3, lines 7-9. Applicants respectfully traverse. *Singer* instead discloses that a filtering program locally generates a customized, filtered log file. *Singer* ¶¶0051]. The filtering program also periodically deletes outdated records in the filtered log file with their related data and files *Singer* ¶¶0052]. In the method of *Singer*, the Collection Server then starts a Collection Program that locally compresses the filtered log files (using WinZip V 6.3) and transfers these zipped files to the Analysis Server. *Singer* ¶¶0054]. Hence, *Singer* discloses the collection, transfer, and management of filtered log files in a central location. However, there is no language in *Singer* regarding closing or reading the current log file, or generating a database file in response to said log file. Thus, *Singer* does not disclose the limitations of claims 2, 9, and 16 and thus *Singer* does not anticipate claims 2, 9, and 16. M.P.E.P. §2131.

Applicants further assert that *Singer* does not disclose "starting a next log file" as recited in the method of claim 3, the computer program product of claim 10, and the data processing system of claim 17. The Examiner cites ¶¶0065] of *Singer* as disclosing the above-cited claim limitation. See Office Action from 11/30/2005, p. 3, line 12. Applicants respectfully traverse and assert that *Singer* instead discloses locally creating filtered log files and periodically deleting outdated records and

related data from the filtered log files. The cited passage by the Examiner discloses repetition of the entire analysis method in *Singer* for each server, and thus is irrelevant to claims 3, 10, and 17. There is no language in the cited passage that discloses starting a next log file. Thus, *Singer* does not disclose the limitations of claims 3, 10, and 17 and thus *Singer* does not anticipate claims 3, 10, and 17. M.P.E.P. §2131.

Applicants further assert that *Singer* clearly does not disclose "closing a current log file", "reading said log file", "generating said database file" and does not disclose that these steps "are performed by said log file parser" as recited in the method of claim 4, the computer program product of claim 11, and the data processing system of claim 18. The Examiner cites ¶[0017] of *Singer* as disclosing the above-cited claim limitations. See Office Action from 11/30/2005, p. 3, lines 13-16. Applicants respectfully traverse. *Singer* teaches a method clearly distinct from the above-cited claim limitations, "the collection process collects the filtered usage records from all the servers on the system, further processes the usage data, and then transfers the data to the Analysis Server where the data is loaded into a relational database." *Singer* ¶[0017]. Hence, *Singer* discloses the collection, transfer, and management of filtered log files and discloses generating a database file after the usage data have been retrieved in filtered form. There is no language in *Singer* regarding closing or reading the current log file locally, i.e. on each server. Neither does *Singer* mention generating a database file with a file parser on each server, as is clearly recited in the parent claims 1, 8, and 15, respectively. Thus, *Singer* does not disclose the limitations of claims 4, 11, and 18 and thus *Singer* does not anticipate claims 4, 11, and 18. M.P.E.P. §2131.

Applicants further assert that *Singer* does not disclose "launching a log file browser" and "retrieving usage information from a database file" as recited in the method of claim 5, the computer program product of claim 12, and the data processing system of claim 19. The Examiner cites ¶[0065] of *Singer* as disclosing

the above-cited claim limitation. *See* Office Action from 11/30/2005, p. 4, line 3. Applicants respectfully traverse and assert that *Singer* instead merely discloses repeating the entire analysis method for each server. *Singer* does disclose a "system for collecting, filtering, analyzing and reporting web server usage data" *Singer* ¶[0008], and a process by which "web server usage data can be collected from multiple servers" *Singer* ¶[0009]. However, the collection process of *Singer* does not contain the elements in claims 1, 8, and 15 (as restated in claims 5, 12, and 19, respectively) of "starting a log file parser" and "generating a database file by said log file parser." The method disclosed by *Singer* begins with a pre-installed filtering program on each web server for locally generating a customized, filtered log file. *Singer* ¶[0051]. In the method of *Singer*, the Collection Server then starts a Collection Program that locally compresses the filtered log files (using WinZip V 6.3) and transfers these zipped files to the Analysis Server. *Singer* ¶[0054]. Hence, *Singer* discloses the collection, transfer, and management of filtered log files. There is no language in *Singer* regarding generating a database file on each server by a log file parser. Thus, *Singer* does not disclose the limitations of claims 5, 12, and 19 and thus *Singer* does not anticipate claims 5, 12, and 19. M.P.E.P. §2131.

Applicants further assert that *Singer* does not disclose steps that are "performed by a shell script" as recited in the method of claim 6, the computer program product of claim 13, and the data processing system of claim 20. The Examiner cites ¶[0100-0103] of *Singer* as disclosing the above-cited claim limitation. *See* Office Action from 11/30/2005, p. 4, lines 8-9. Applicants respectfully traverse. *Singer* teaches a method clearly distinct from the above-cited claim limitations, "[i]ninstalling the SQL database on the Analysis Server consists of three database scripts which must be run in the following order: (1) Create Tables Script, (2) Create Views Script, and (3) Stored Procedures Script." *Singer* ¶[0100]. Hence, *Singer* discloses initializing a database environment by running SQL (Structured Query Language) scripts. There is no language in *Singer* regarding running shell scripts, i.e.

bsh, bash, ksh, csh, tsch, as defined in the disclosure (Detailed Description, p. 8, lines 3-5). Thus, *Singer* does not disclose the limitations of claims 6, 13, and 20 and thus *Singer* does not anticipate claims 6, 13, and 20. M.P.E.P. §2131.

Applicants further assert that *Singer* does not disclose log files that "are maintained by a directory server" as recited in the method of claim 7, the computer program product of claim 14, and the data processing system of claim 21. The Examiner cites ¶[0043] of *Singer* as disclosing the above-cited claim limitation. See Office Action from 11/30/2005, p. 4, lines 12-13. Applicants respectfully traverse. *Singer* teaches a method clearly distinct from the above-cited claim limitations, "[t]he three types of computer systems which are utilized by the invention include: (1) web servers, (2) Collection Server, and (3) an Analysis Server." *Singer* ¶[0043]. Hence, *Singer* discloses and claims a web site usage data collection, analysis and reporting system wherein the log files are maintained by web servers. *Singer* ¶[0041, 0002-0007, 0008, 0009, 0010, 0048, 0050-0055, 0065, ABSTRACT]. There is no language in *Singer* regarding directory servers, or any other kind of network server, except for web servers. Directory servers are well appreciated in the art as being distinct and separate from web servers.

**directory server** - a network server that provides a directory, or naming service. (<http://www.techweb.com/encyclopedia/defineterm.html?term=directory+server&x=20&y=7>)

**directory service** - a directory service organizes computerized content and runs on a directory server computer. It is not to be confused with the directory itself, which is the database that holds the information about objects that are to be managed by the directory service. The directory service is the interface to the directory and provides access to the data that is contained in that directory. It acts as a central authority that can securely authenticate resources and manage identities and relationships between them. ([http://en.wikipedia.org/wiki/Directory\\_server](http://en.wikipedia.org/wiki/Directory_server))

Hence, a method that operates solely on a web server does not anticipate and is patentably distinct from a method that operates on a directory server. Thus, *Singer* does not disclose the limitations of claims 7, 14, and 21, and thus *Singer* does not anticipate claims 7, 14, and 21. M.P.E.P. §2131.



As a result of the foregoing, Applicants respectfully assert that not each and every claim limitation was found within the cited prior art reference, and thus claims 1-21 are not anticipated by *Singer*.

### III. CONCLUSION

As a result of the foregoing, it is asserted by Applicants that claims 1-21 in the Application are in condition for allowance, and Applicants respectfully request an allowance of such claims. Applicants respectfully request that the Examiner call Applicants' attorney at the below listed number if the Examiner believes that such a discussion would be helpful in resolving any remaining issues.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorney for Applicants

By: 

Kelly K. Kordzik  
Reg. No. 36,571

P.O. Box 50784  
Dallas, TX 75201  
(512)370-2832